

# **Vehicle Control System Development**

Mitch Gavrilash  
Coastal Systems Station, Dalgren Division  
Naval Surface Warfare Center  
6703 West Highway 98  
Panama City, FL 32407-7001  
Phone: (850) 235-5886    Fax: (850) 234-4867    E-mail: [mitch@ieee.org](mailto:mitch@ieee.org)

Document #: N0001400WX20029  
<http://www.ncsc.navy.mil/>

## **LONG-TERM GOAL**

Develop optimized search strategies and corresponding vehicle control systems for swarms of autonomous vehicles performing missions in the littoral surf zone.

## **OBJECTIVES**

Develop layered behavior based control system in simulations and actual hardware in direct support of the Surf Zone Reconnaissance and Demonstration Plan.

## **APPROACH**

Develop autonomous vehicle behaviors, command structure, communication protocols for control, vehicle report-back, and map making/representation. Implement in simulations, firmware and test/validate experimentally.

Augment, as needed, the Distributed Underwater Network Evaluator (DUNE) for statistical analysis, control systems development and robotic swarm visualization. Define the requirements for the operator control station interface. Pursue compatibility with other simulations and existing C4I infrastructure.

## **WORK COMPLETED**

Defined a command set and a communication protocol for use over a narrow band, time multiplexed channel, to control a swarm of crawling vehicles (October - November 1999). Presented the approach at the Office of Naval Research (ONR) Surf Zone/Very Shallow Water (SZ/VSW) 2<sup>nd</sup> Program Review/Workshop (December 1999).

Table 1 summarizes the Coastal Systems Station (CSS) Vehicle Command Set Draft 2.0 that includes Foster-Miller Inc. (FMI) Lemmings vehicle's five Remotely Operated Vehicle (ROV) commands (f, b, l, r and s), while attempting to maintain a one-to-one correspondence with some of Autonomous Underwater Systems Institute (AUSI) vocabulary.

Convention is that lower case letter commands do not have arguments and are used for remotely operating the vehicles as well as providing access to internal variables. Capital letters are commands with arguments for elementary control and for setting vehicle parameters. We believe this command set is sufficient for controlling an autonomous vehicle and recognize that there is some command redundancy.

**Table 1. CSS Command Set (Draft 2.0)**

Elementary	ROV / Diagnostics / Primitives	
	<i>Command</i>	<i>Reply</i>
A(djust relative timer) # (seconds)	a(ccelerate)	a # (nominal speed)
B(ack-up) # meters	b(ack-up)	b
C(orrection) # RTCM String (bytes)	c(ordinates)	c GPS String
D(estination) sets global X#,Y#, Z#	d(ecelerate)	d # (nominal speed)
E(XECUTE CONVENTION) #	e(nergy)	e # (percent)
F(orward) # meters	f(orward)	f
G(roup) # (assign)	g(roup)	g # (currently assigned to)
H(eading) # adjust vehicle orientation	h(eading)	h # (degrees)
I(mage) # (bytes)	i(motor currents)	i # # (left, right Amperes)
J(og) # diagnostic maneuver type	j(og)	j #
K # set heartbeat interval in seconds	k(eep heartbeat going)	k # (stop timer setting)
L(eft) # (relative degrees)	l(eft)	l
M(OVE) V #, X #, Y#, Z #	m(achine)	m # (id)
N(avigation bounds) e.g., $Z < aX + b$	n(avigation boundary)	n # nav. bound or obstacle
O(rigin) #, #, # sets origin coordinates	o(rientation)	roll, pitch, yaw
P(osition) X#, Y#, Z# (relative)	p(osition) rel.	x, y, z
Q(uey) # report type	q(quiet) mode	q # toggle verbose flag
R(ight) # (relative degrees)	r(ight)	r
S(top) all stop	s(top motors)	s
T(RANSIT) range #, bearing #	t(ime)	t # (seconds)
U(nit) # access subsystem	u(nknown object type)	u # (object type)
V(elocity) # # set	v(elocity)	Vx, Vy, Vz, V
W(aypoint) index #, X #, Y#, Z #	w(heel counts)	w # # (left, right count)
X # pos. in North latitude direction	x	x # (relative meters North)
Y # depth/altitude, zero at sea level	y	y # (depth)
Z # pos. in the East longitude	z	z # (relative meters East)

Microbehaviors are proportional-integral-differential (PID) loops or finite state machines comprised mainly of elementary functions. Example microbehaviors are direction tracking, obstacle avoidance, sensing strategies and search patterns. A scripting mechanism based on an interactive language such as LOGO or FORTH can provide conditional branching, loops, arithmetic and logic operations for microbehavior development.

We ported the Lemmings vehicle control system from Motorola 68C16 to AMD 188 ES based family of microcontroller boards to pursue compatibility with the new FMI Tactical Adaptable Robot (TAR) class vehicles. We embedded a mini-FORTH command interpreter to support mission scripting in terms of

elementary control commands (January-February 2000). FORTH was selected for execution speed, since new words are only interpreted once, then compiled and placed in an internal dictionary. We implemented a PID control loop that effectively tracks direction in natural environments. The compass-based PID loop was tuned on a buggy steered test bed platform (March 2000, Figure 1) and subsequently refined for the TAR vehicle (April 2000, Figure 2).

The classic first order discrete time equivalent of the derivative term has been implemented by a higher order difference equation (3) to overcome susceptibility to noise.

$$Err(i) = Intended\_Heading - Compass \quad (1)$$

$$Err\_Integral = Err\_Integral + Err(i) \quad (2)$$

$$Delta\_Err = ( Err(i) + 3 * Err(i-1) - 3 * Err(i-2) - Err(i-3) ) / 6 \quad (3)$$

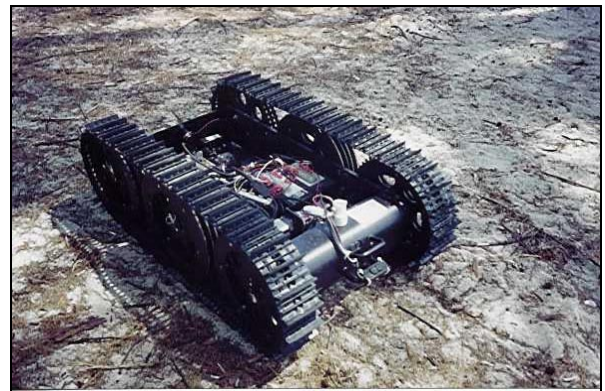
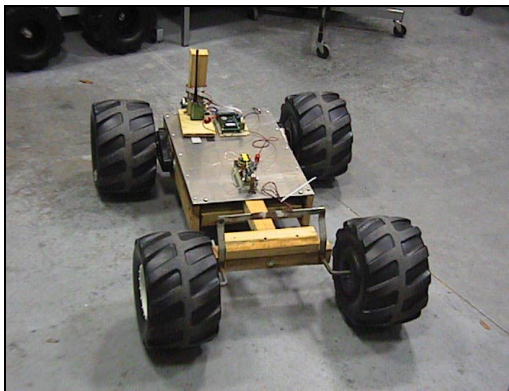
$$PID\_Output = Kpro * Err(i) + time\_intv * Kint * Err\_Integral + (Kdif / time\_intv) * Delta\_Err \quad (4)$$

Anti-reset-windup check done here.

$$Speed\_Left = Nominal\_Speed + PID\_Output \quad (5)$$

$$Speed\_Right = Nominal\_Speed - PID\_Output \quad (6)$$

The  $Kpro$ ,  $Kint$  and  $Kdif$  gain factors were accessed via the command interpreter and adjusted experimentally. For the tracked vehicle, only the proportional and integral (PI) terms are used during forward movement. Setting nominal speed to zero and adding the differential term to the PID controller will execute skid turns. Even without gyro inertial sensing, this is one way of orienting the machine close to the intended heading in spite of compass lag and vehicle moment of inertia.



**Figures 1 & 2. Buggy Steered and Land TAR Vehicles**

We integrated the vehicle control system with navigation subsystems and the position estimation software. Waypoint navigation was tested on land using dead reckoning and DGPS baseline (05/00).

Underwater autonomous waypoint navigation using the CSS developed Lemmings Acoustic Navigation System (LANS) acoustic baseline was achieved in the West Bay. The completed system, which included a JAVA implementation of the operator control station (OCS), was demonstrated in the

Atlantic Ocean at the South Florida Ocean Measuring Center (SFOMC) site during the Fleet Battlelab Experiment-Hotel (FBE-H) rehearsal (June 13, 2000). Figures 3 and 4 show the completed system, on the beach and underwater.

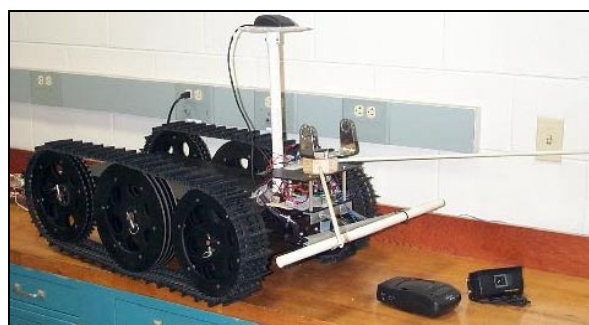


**Figures 3 & 4. Launch and Underwater Autonomous Waypoint Navigation**

The mission consisted of seven individual waypoints, followed by a scripted path comprised of a sequence of four relative destinations expressed in range and bearing format. We defined a search path by entering at the command prompt the following: “: *path 20 90 T 4 180 T 20 270 T 4 180 T*;”. Two additional waypoints returned the vehicle to shore.

Lawnmower pattern could be expressed by repeating *path* N times: “: *search N 0 DO path LOOP*;”. With the addition of obstacle detection electronics (Figure 5), random walk and obstacle avoidance were implemented via the command interpreter (July-August, 2000). Example safe wandering script:

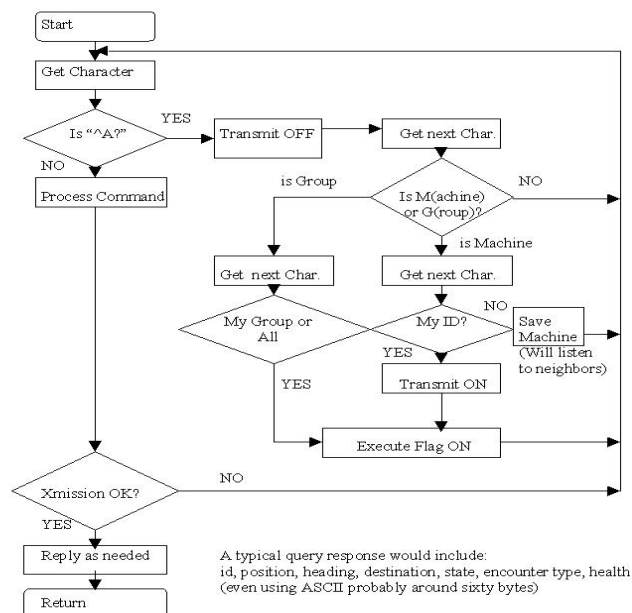
```
: rmdm_dist    maxdist GET RND mindist GET + F ;    ( move forward a random distance )
: rmdm_turn    maxturn GET RND H ;                  ( execute a random angle turn )
: back         backdist GET B rmdm_turn ;           ( back-up followed by random turn )
: back_flag    collision GET bounds GET OR ;         ( determine if need to back-up )
               ( execute random walk in a bounded area until relative timer exceeds 600 seconds )
: walk BEGIN rmdm_dist back_flag 1 = IF back ELSE rmdm_turn ENDIF t GET 600 > UNTIL ;
```



**Figure 5. Obstacle Detection Hardware: Bumper, IR Detector and Capacitive Whiskers**

For controlling more than one vehicle at a time, the OCS provides a terminal window for human to vehicle swarm interaction. The vehicles can be controlled individually or assigned to and addressed in

groups. The flowchart below depicts the 'data link media access layer' based on NMI's Easy-A protocol.



## RESULTS

We refer to the combination of the above mentioned command set, the interpreter and the asynchronous communication scheme as ROBOTALK. Essentially, ROBOTALK provides the means to remotely program, command and control a swarm of autonomous robots. Its immediate use is to support the multiple-robot search and area coverage tests, scheduled later this fall, and for the rest of FY01, to interactively develop sensing strategies that utilize the vehicle mobility to maximize sensor effectiveness in the harsh, surf zone environment.

## IMPACT/APPLICATION

The rationale behind ROBOTALK was dictated by the need to dynamically develop a vehicle behavior library suitable for surf zone MCM operations. This effort complements common control language (CCL) work at AUSI, Penn State and NUWC/USC in that it focuses precisely on the level of vehicle hardware abstraction. Interfaces to higher level of control can be established through scripting.

## TRANSITIONS

The control system code has been made available to FMI. The scripting mechanism can facilitate the implementation of the JAUGS communication protocol required for other DOD funded projects.

## RELATED PROJECTS

EOD Technology Division, Indian Head, MD: Basic Unexploded Ordnance Gathering System (BUGS) Program.

## REFERENCES

L. Brodie, 1984. Starting Forth, Prentice-Hall.

N. Smith, 1996. Write Your Own Programming Language in C++” (2<sup>nd</sup> Edition), Worlware Publishing.

D.W. Gage, 1995. Development and Command-Control Tools for Many-Robot Systems, *Proceedings of SPIE Microrobotics and Micromechanical Systems*, Philadelphia PA, 23-24 October, volume 2593, pp 121-129.

MAXDDE User Manual, Version 1.1, NewMicros Inc., 1995.

C. Duarte, 2000. Port Arbitrated Behavior Common Control Language Outline, May.

Rick Komerska, Steve Chappell, Liang Peng, and Dick Blidberg, 1999. Generic Behaviors as Interface for a Standard AUV Command & Monitoring Language, AUSI Working Draft Version A3, September.

S. Phoha, 1999. A Proposal for a Generic Behavior Message-Passing Language for Collaborative Missions of Heterogeneous Autonomous Underwater Vehicles, PSU/ARL Memo to AOSN Community.

Tactical Control System Segment to Air Vehicle Standard Segment Interface Design Description, NSWCCD TCS-229, August 1997.

Joint Architecture for Unmanned Ground Systems (JAUGS), Draft Version 1.0, Unmanned Ground Vehicles/Systems Joint Project Office, AMSAM-DSA-UG.

D.W. Gage, 1994. Cost-Optimization of Many-Robot Systems, *Proceedings of SPIE Mobile Robots IX*, Boston MA, 2-4 November, volume 2352, pp 260-266.

Control EG, Quinn-Curtis, 1988.

C. Nachtigal, 1990. Instrumentation and Control: Fundamentals and Applications, Wiley-Interscience.